A SYSTEM ARCHITECTURE FOR A PLANETARY ROVER

D.B. Smith and J.R. Matijevic
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive
M/S 198-105 and 303-308, Pasadena, California 91109

ABSTRACT

Each planetary mission requires a complex space vehicle which integrates several functions to accomplish the mission and science objectives. A Mars Rover is one of these vehicles, and extends the normal spacecraft functionality with two additional functions: surface mobility and sample acquisition.

This paper assembles all functions into a hierarchical and structured format to understand the complexities of interactions between functions during different mission times. It can graphically show data flow between functions, and most importantly, the necessary control flow to avoid unambiguous results.

Diagrams are presented organizing the functions into a structured, block format where each block represents a major function at the system level. As such, there are six (6) blocks representing Telecomm, Power, Thermal, Science, Mobility and Sampling under a supervisory block called Data Management/Executive. Each block is a simple collection of state machines arranged into a hierarchical order very close to the NASREM model for Telerobotics.

Each layer within a block represents a level of control for a set of state machines that do the three primary interface functions: Command, Telemetry, Fault Protection. This latter function is expanded to include automatic reactions to the environment as well as internal faults.

Lastly, diagrams are presented that trace the system operations involved in moving from site to site after site selection. The diagrams clearly illustrate both the data and control flows. They also illustrate inter-block data transfers and a hierarchical approach to fault protection. This systems architecture can be used to determine functional requirements, interface specifications and be used as a mechanism for grouping subsystems (i.e., collecting groups of machines ,or blocks consistent with good and testable implementations).

1. INTRODUCTION

The history of operational planetary rovers begins with the USSR Lunokhod-1 mission on the moon, a nearly one year mission beginning in November, 1970. The Lunokhods were interactively controlled, starting and stopping according to planned sequences created by a ground mission team receiving TV images.

An advancement on this design principle is the JPL's Computer-Aided Remote Driving or (CARD) system implemented on a six-wheeled test vehicle. This system was developed under sponsorship of the U.S. Army Tank-Automotive command and demonstrated the capability of on-board execution of a human operator selected path drawn on a frozen image of the local terrain (Reference 8).

Current rover concepts vary from advanced concepts of this design (CARD) to highly automated vehicles performing automatic collision avoidance (for example, JPL's Semiautonomous Navigation or (SAN), Reference 1). Unlike the Lunokhods, concepts under study for a Mars rover mission (Reference 9) must accommodate in-situ sampling, very much like an automated geologist. This complexity dictates a flexible systems architecture invariant to different levels of automation.

Because of the nature of the mission, the architecture must combine the functionality of planetary spacecraft with the additional functions of mobility and sampling. Inclusion of these last two major functions dramatically expands traditionally layered architectural structures for spacecraft systems.

An architecture which incorporates a mobility function, must provide a structure for accommodating simple to complex walkers as well as the more traditional wheeled carriage vehicles. One of the simplest walker concepts is Brooks' 1 kg rover consisting of fifty-six (56) state machines cycled individually in a particular pattern to produce a "gait" (Reference 2). A more complex walker concept is the Carnegie Mellon University "ambler" (Reference 3). In between may be considered the elegant "beam" walker concept of Martin Marietta Corporation (Reference 4). In each case, a multi-layered architectural structure is suggested where control of individual walker link motors or wheels are coordinated at higher levels to produce the desired motions.

The addition of a sampling function introduces the added complexity of control of manipulations. As recent studies suggest (see for example the architectural concepts for telerobotics proposed for the Goddard Space Flight Center (GSFC) Flight Telerobotics Servicer (FTS) (Reference 10) and for the NASA/OAST Telerobot Testbed (Reference 11)) multi-layered control structures are required to coordinate manipulator and end-effector/tool link controllers. Precursor missions such as a Mars Sample return concept require a high level of automation of the sampling function. But later manned missions, with associated human interaction with the rover, require consideration of a range of supervisory control options for the sampling function. Consequently, human interactions must be smoothly integrated into a control architecture for the rover.

In this paper, a single architectural concept integrating all of the above is proposed for the general class of planetary rover concepts in the 1990's. The architecture is a loosely coupled, state-machine concept incorporating the hierarchical control concepts of NASREM (NASA/NBS Standard Reference Model for Telerobot Control System Architectures, Reference 5).

The following describes the architectural concept and provides a mobility scenario, tracing the sequential execution of several functions within the control layers of the architecture. A sampling scenario has also been done for completeness but is not presented here for the sake of brevity. Both scenarios validate the architecture's flexibility and accuracy.

Lastly, some final thoughts are presented on the uses of the architecture. These observations should apply to any good architecture, not just this one. For example, a typical systems design task is the mapping of the system functions into an implementation by a number of subsystems, defining boundary interfaces at simple junctions (e.g., functional levels). This architecture naturally decomposes into the standard subsystems for planetary spacecraft and provides a structure for the evaluation of alternate decompositions.

2. PLANETARY MISSIONS

Before discussing details of the architecture, we introduce some of the basic functions and subsystems of any planetary spacecraft. A Mars Rover, after all, is at least a planetary spacecraft and much more.

Basic subsystems of any planetary spacecraft include Telecommunication, Power, Thermal, Attitude Control, and Science. Common to these subsystems are three activities: receipt and/or processing of commands, telemetry output and fault protection. Due to the delays in transmission to a ground station, these subsystems must at a minimum fail safe, and the spacecraft as a whole must fail operational, albeit in a degraded mode. Consequently, each subsystem must detect errors and take local action such as the use of a redundant string. Once the error correction is accomplished the subsystem notifies a command and control subsystem (if available) of the configuration change. Of course, this information is passed on to ground command as soon as possible for evaluation and further corrective action.

No planetary spacecraft has been designed to be fully autonomous. Instead, these spacecraft are semi-autonomous (in the sense of the above discussion), relying heavily on ground control for mission commanding, analysis, and planning. Therefore, any general architecture for spacecraft control will include a significant ground segment. Functions may move from the ground to the spacecraft if more risk is assumed or capability made available. An example of this enabled the extended Viking mission. After the primary mission and with suitable confidence in the spacecraft capability, the Viking program reduced its operational ground staff significantly, by reprogramming the flight computers on the orbiters with automatic routines for fault protection.

The MRSR (Mars Rover and Sample Return) mission contains all the subsystems and functions of a planetary spacecraft except for the classical Attitude Control subsystem. Attitude of the Rover is monitored as a part of mobility but not controlled until some limit is exceeded. For example, an inclinometer may detect a dangerous tipping-over attitude which requires system action for correction.

The additional rover-specific subsystems are Mobility and Sampling. Mobility contains the local navigation function vital to semi-autonomous traverse. This local navigation function is very important, requiring interactions among other functions. Some architectures for a rover represent only this viewpoint. For example, Kovtunenko (Reference 6) represents the local navigation function as the main interface to the Earth and central to all other functions within the rover. This view is useful for describing control principles, but neglects other major rover functions.

3. MRSR MISSION

A MRSR mission has many variations (including some without a separate rover vehicle). The following summarizes main mission and operational requirements for a rover in the MRSR mission.

The rover greatly extends the number and types of samples which can be returned from Mars (Reference 7). A nominal rover operation entails a sequence of local traversing segments and sampling, constrained by on-board resources and ground interactions. Prolonged or extensive decision time by the ground operations can severely limit the rover's integrated range (from a goal of 40 km-100 km).

After each traverse, a sample may or may not be collected, depending on science value as determined by an on-board evaluation or ground mission team decision. Some strategies allow collection of samples without in situ discrimination. After a collecting tour, the rover locates the MAV (Mars Ascent Vehicle) for a rendezvous and sample hand-over. (The MAV may have resulted from an integrated rover or separate launch).

For the purpose of this paper, consider that each traverse segment is using semi-autonomous navigation. Figure 3-1 is a functional diagram of the activities. At a certain point, the ground up-links a topographic map from a ground-based Global Route Planner. This map contains a first-order ground swath for the rover's traverses and a designated path(s) avoiding obstacles and dead-ends that remains scientifically interesting.

The rover takes a panoramic view of the local scene using stereo cameras, laser scanning, or structured light, linked to machine vision and image processing functions. The rover computes a local map from the processed view and matches this map to the local portion of the global topographic map sent from Earth. Using the results of the match, constraints of previous rover positions and output of any other navigational devices, the rover determines an accurate position of itself. A revised (fused) map is formed from the two sources (ground/global and rover/local) to produce a high resolution map in the vicinity. A new path then is computed via simulation, revising the approximate path sent from Earth by including (and thus excluding from the path) small obstacles not detected by the low resolution images of the global map. The original global resolution is required to be accurate to 1 meter. The fused map could be accurate to 10cm. The rover then traverses the path.

In planning the route, the simulation of the traverse of the path is used to compute slope changes and tilt expectations. These are used in a predictive way to set limits on inclinometer and local proximity sensors. These predictions control the rover's reflex actions in the event an errant path is followed.

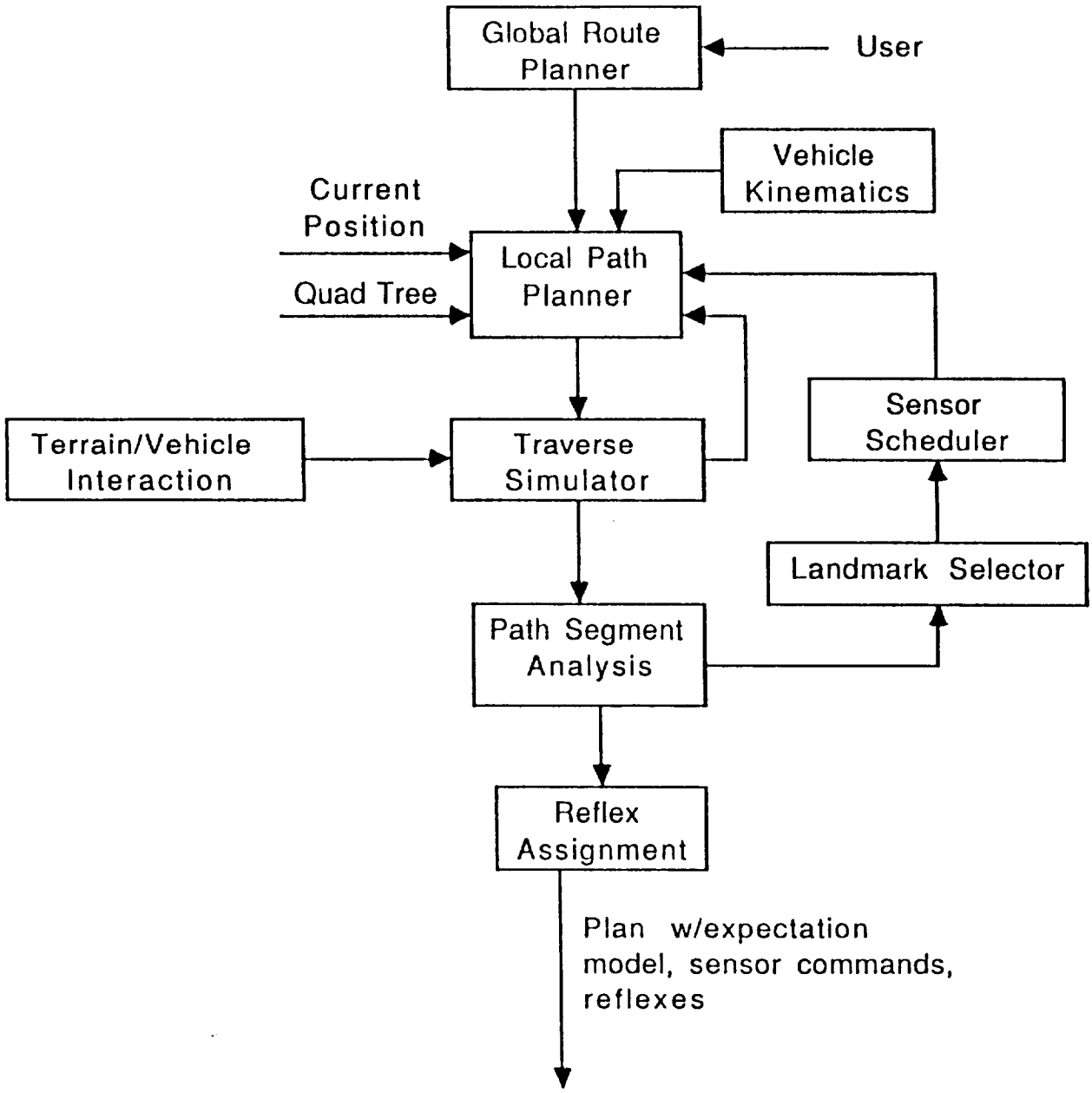

Global Route Planner
User
Vehicle Kinematics
Current Position
Quad Tree
Local Path Planner
Sensor Scheduler
Terrain/Vehicle Interaction
Traverse Simulator
Landmark Selector
Path Segment Analysis
Reflex Assignment
Plan w/expectation model, sensor commands, reflexes

FIGURE 3-1

If enough computational resources exist, the cycle of global map up-link(as needed) - fused map development-path planning - traverse can be repeated every few minutes, for a resultant average speed of 10cm/sec (see Reference 1). Typically a local path is computed for about thirty (30) meters, with the rover pausing every ten (10) meters or so to re-evaluate its condition.

4. NASREM

NASREM is a standard telerobotic architecture now adopted by the FTS, its contractors and many others associated with the Space Station program (Reference 5).

The NASREM is a modular, hierarchical conceptual architecture for telerobotic system control (see Figure 4-1). The main feature of NASREM are six layers of control:

(0) the world: hardware elements, being controlled by the telerobot and the environment of operation
(1) the servo level: coordinates are transformed and outputs servo the arms/end-effectors
(2) the primitive level: telerobot dynamics are computed and coordinated arm commands issued
(3) the E-move level: obstacles (including those inherent to arm operation) are observed and commands issued to avoid them
(4) the task level: tasks to be performed on objects are transformed into movements of effectors
(5) the service bay level: task on groups of objects in the vicinity of the telerobot are sequenced and scheduled
(6) the mission level: objects are collected into groups, resources are assigned between telerobots and parts/tools are routed and scheduled

Each NASREM layer is partitioned into three modules: sensory processing, world modeling and task decomposition. Additional features include a global memory to support the flow of information and coordination between levels in the hierarchy and an operator interface to support operator input and display capabilities at all levels of the hierarchy.

Since the control levels are well ordered, unambiguous commands flow from the top of the hierarchy down to the lowest level of the servomechanism. This is role of the task decomposition elements of the architecture. At each level in task decomposition, a job assignment manager partitions task commands into distinct jobs to be performed by one or more planner/executor modules. As such, each planner is responsible for decomposing a job command into a temporal sequence of planned subtasks. The executor evaluates the sequence prior to execution.

Data or status flows in reverse from the lowest levels of the hierarchy to the top levels. This data is available from three sources. The data may be fed back from one level to the next through the hierarchy in task decomposition. Alternately, depending on the use/need, the data may be read from the global memory. This global memory is a data base where knowledge about the state of the task space, task environment and internal state of control system is stored. Each layer of the hierarchy and all processing modules within a layer contribute to global memory. Lastly, data may be received through the interaction of the telerobot system with the environment. This data is obtained through the sensory processing elements of the architecture. Sensor information is read and processed in a hierarchy which allows the system to recognize patterns, detect events, filter and integrate information over space and time.

The processing of the data or status is performed using models of the effectors, sensors or environment of the telerobot. This is the role of the world model elements of the architecture. These models include estimates and evaluations of the history, current state and possible future state of the telerobot system and task space. These models help maintain the data in global memory, offering confidence levels/statistics of model predictors and sensory observation.

The last elements of the architecture are contained in the operator interface. The functions in these elements enable interface to each level of the hierarchy. The operator can enter the control flow to monitor a process, insert information, interrupt automatic operation, take over, and apply human intelligence to the processing. Feedback ranges from force reflection at the lowest levels to displays for interactive scheduling at the highest levels.

In guiding functional partitioning, this architecture constrains functions to a given layer by processing rate or bandwidth. At the lowest levels, the processing is severely time dependent in stabilizing servo control loops. As such, rates of execution may be as high as 1000 times per second (or 1000Hz). At each higher level the rates decrease generally by a factor of 10 or more.

In implementing the architecture, the available technology and operational priorities dictate additional constraints. The processing load at the lowest levels lead to optimization of communication paths. Thus, the data in global memory which serves inter-process communication at these levels may be kept separate from data at other levels, with access prioritized by need for the control loops. The need for verification of command sequences at the higher levels of the architecture lead to the inclusion of simplified (though correct) models of specific hardware in the world models at these levels.

5. ROVER SYSTEM ARCHITECTURE

The system architecture for the rover discussed in this section is based on the NASREM model. We generalize this model in two ways. We unify treatment of rover subsystems and specific functions such as command and telemetry by modeling each in a two-dimensional NASREM architecture. We collect the elements in a four dimensional array space, allowing a simple mechanism for sorting elements by like function and processing. In addition, we generalize the definition of the intermediate layers (Levels 1 through 3) of the architecture allowing expansion to such functions as mobility and telecommunications, while maintaining the spirit of the definition of these levels for a telerobot.

In considering then the definition of a NASREM model for use in a rover system, we utilize the concept of small state machines introduced by Brooks (Reference 2). Level 1 functions can simply be interpreted as those state machines which implement the settings of dynamical systems or the states to be controlled. In the Brooks' walker, these states are the different positions of the legs. In the control of a robotic arm the states are the various settings of the joints of the arm. Level 2 functions set permissible states as represented by a control law, constrained by various dynamic and kinematic models. At the next level, a sequence or function of permissible states implements a subtask or operational process for the rover. At this level, for example, the movement of a walker along a path is a sequential execution of repositions, with each reposition itself a

NASREM - NASA/NBS STANDARD REFERENCE MODEL

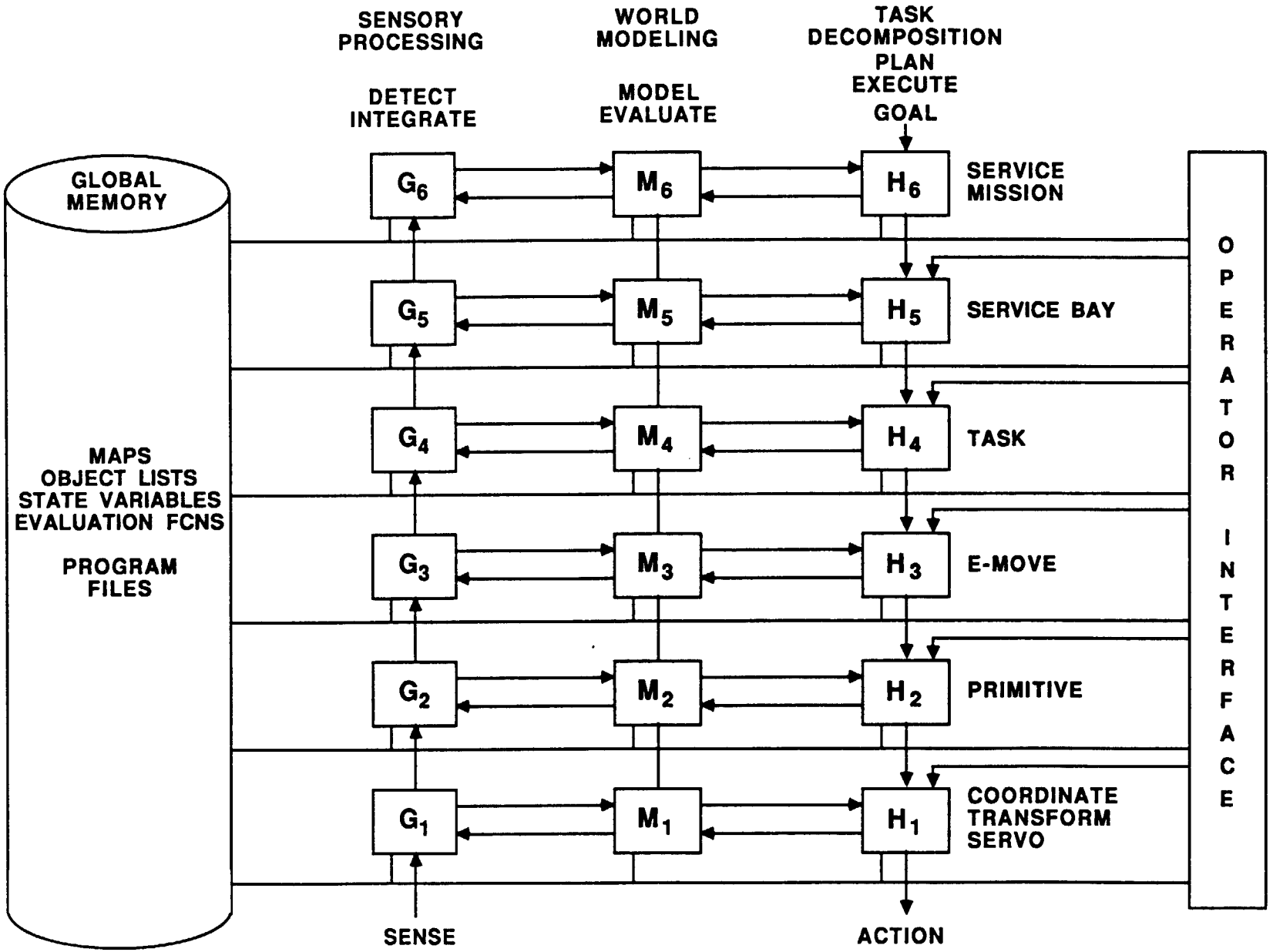


FIGURE 4-1

sequence of leg motions in a controlled, coordinated (or gaited) move. For an arm, a cartesian-space referenced path is achieved by moving the arm joints to achieve a series of end-points. Task planning at the highest level generates collections of sequences which accomplish a task. For example, movement of an arm or rover from point A to B is achieved by the execution of a set of commanded path segments.

Figure 5-1 is a generalized version of the NASREM model where this new state-definition terminology is applied. In another change from Figure 4-1, we have re-ordered the columns adding a slice so that a distinction is made between world models of the environment being sensed and world models of devices being commanded. The global data base becomes the middle slice and contains the constants and parameters of the world models as well as the data comprising the state of the overall system.

To the functional processing machines, we have added a third dimension to the architecture: a stack of command and a stack of telemetry machines. Commands flow down and telemetry flows up. In between is a separate stack of fault protection machines which can be excited and alter commands and telemetry at any given time. Fault protection machines execute as a result of a detected error read through telemetry. They implement recovery actions through commanding of the functional processing elements to assume a new state. As an example for a mobility subsystem, an inclinometer may sense an excess tilt angle. A corresponding fault protection machine will detect the error based on the telemetry and then act by issuing a command to cease forward motion. In addition, an appropriate routine may be executed such as carefully retracing the last series of commands until acceptable tilt angles are achieved (i.e., back-up). Once completed and the vehicle is safed, a route replanning will be commanded.

Notice that in the above example command and telemetry machines interact with the fault protection machines. The fault protection machine orchestrates the actions of the functional processing machine(s) implementing recovery. In detail then for the above example, an inclinometer at Level 1 registers excessive tilt causing a Level 1 fault machine to interrupt commands from Level 2 thereby halting the system. Telemetry is then sent to the Level 4 fault machines which have been receiving some subset of the last successfully commanded states. The Level 4 fault machine then calls for the task execution of a traverse back to a point along the route. Commands then flow normally downward until this previous state is achieved. When this happens, system control is then passed back to the task planner which knows its path plan has been altered and must replan a new path, which has a prescribed set of greater margins of expected tilt angles along its simulated path.

If any of these actions require more power or other subsystem intervention, then there is data flow between subsystem. Each such subsystem is represented by a block as shown in Figure 5-2. This is the fourth dimension of the architecture.

Returning to the basic functions of a spacecraft, we can now construct the entire model of a Mars Rover (see Figure 5-2). It consists of seven "cubes" in all. The Data Management System is an automated version of the Command Data System, complete with a tasking level. Each element is a state machine, so there are 560 state machines not counting the world elements at the bottom. Notice this is exactly ten (10) times the 56 machines of the insert Brooks' walker. Furthermore, each small machine is considerably more complex.

There are some interesting features of this architecture. For example, the front faces all represent the control flow, while the sides are data flow. This gives one a complete look at the total information system during the design process. Also, it is important to note the information flows between blocks. This is accomplished using the global memory of each face of each block. The four dimensional property of the architecture allows a an arrangement which make communication possible among the global memory (i.e., global memory of the system architecture is a 'block' cross-section of the four dimensional 'cube').

There is much information being exchanged between the Data Management System block and the other subsystem blocks. Also note that data from the ground first appears in the Telecommunications block and then to Data Management. Data Management controls the other blocks through its commanding sequence in order to accomplish a task.

As an example of a planetary spacecraft viewed in this architecture, Voyager's task planning was a function of the ground mission control team. Its Data Management System is the CCS or Command & Control Subsystem. Its command and control functions can be mapped to the lower two (2) levels of the DMS architecture (see Figure 5-3). The mobility function (Figure 5-8) degenerates to a three level Attitude & Control Subsystem. Sampling disappears altogether. In this architecture the Voyager FDS (Flight Data System) is the sum of all the telemetry state machines. The Science block contains the commanding of the science platform, which may be represented in a single level architecture.

The Fault Protection Subsystem for Voyager consisted of automatic recovery routines; little on-board tasking based on sensory information was allowed. Therefore we at once can write down the totality of the Voyager Flight Fault Protection system as the sum of the Fault Protection slices up to level 3. This sum is given in Figure 5-2.

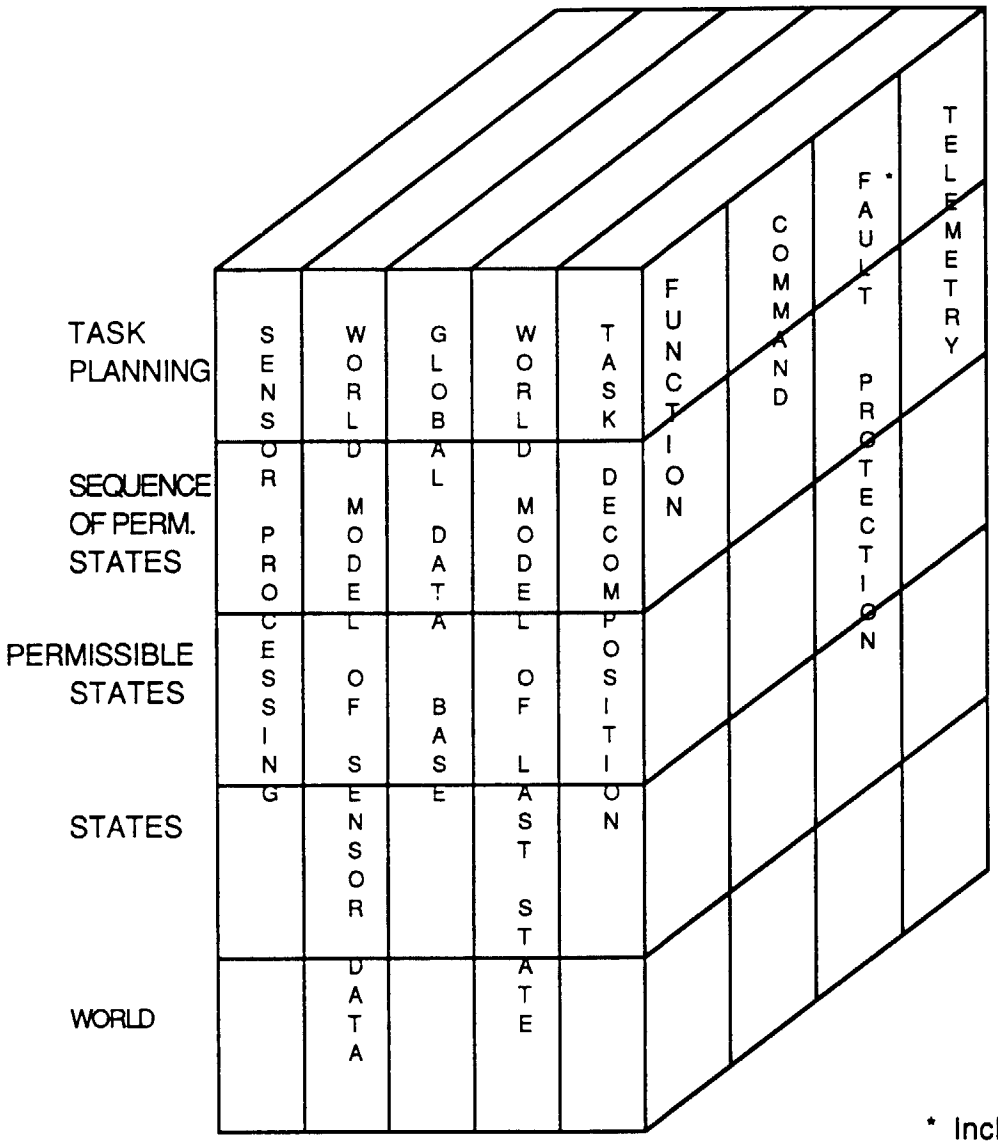
We wish to emphasize that all parts of the blocks exist on Voyager and only certain levels were flown. The remaining levels for Voyager were all on the ground. As spacecraft become more automated, functions and/or levels can be transferred to the spacecraft. These trade-offs are not always implemented since a degree of risk is incurred by this transfer.

The current concept of the Rover has more of these functions on-board than standard planetary spacecraft. In this case, increased on-board autonomy not only reduces ground operations but increases range and (thereby) science measurably.

The remaining Figures 5-3 through 5-9 round out the complete architecture. Notice that some state machines are inoperative, a reflection of today's level of conceptual design. In most cases, however, these state machines are simply not needed (see e.g., Thermal above level 1 in Figure 5-6).

In order to penetrate the design further, the following section will focus on the mobility block which contains the local navigation function. As we mentioned earlier, this is so important to the rover designers that they often view the world as a large mobility block supported by other subsystems.

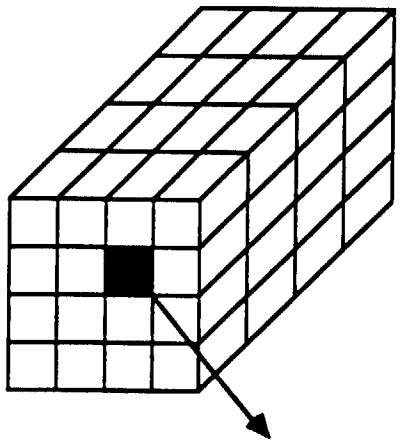
ROVER SYSTEM ARCHITECTURE



* Includes Health and Maintenance

FIGURE 5-1

ROVER SYSTEM ARCHITECTURE

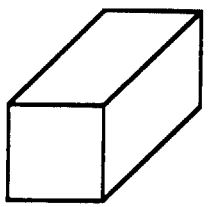


ELEMENT $M_{ij}^{k}(l)$

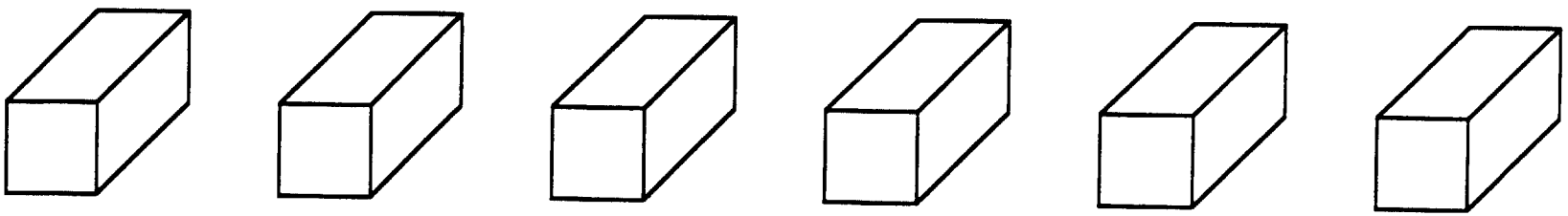
EXAMPLE : (VOYAGER FAULT PROTECTION)

$$FP(3) = \sum_{k=1}^{6} \sum_{i=1}^{5} \sum_{j=1}^{3} M_{ij}^{k}(3)$$

l = 1, ..., 4 = faces
k = 1, ..., 7 = cubes (subsystems)
i = 1, ..., 5 = columns (proc. categories)
j = 0, ..., 4 = rows (hierarchy)



DATA MANAGEMENT (EXECUTIVE)



TELECOMM POWER THERMAL SCIENCE MOBILITY SAMPLING

FIGURE 5-2

DATA MANAGEMENT (EXECUTIVE)

SENSOR PRSING.	(SENSOR) WORLD MODEL	GLOBAL DATA B.	(STATE) WORLD MODEL	TASK DECOMP.	
o Task progress reports	o Task error models	o Sampling locations and constraints o Science tasks	o Simplified task execution models o Task constraint and resource models	o Planner for task execution o Task commands	TASK PLANNING
o s/s operational condition	o s/s operation error models		o Simplified s/s operational models o Simplified s/s constraint and resources models	o Planning for s/s operation o Sequencing of s/s operations	SEQUENCE OF PERMISSIBLE STATES
o s/s state condition	o s/s state error models		o Simplified s/s state model o s/s state constraint and resource models	o Organization of s/s states o s/s state commands	PERMISSIBLE STATES
o Data reports from s/s	o Command acceptance feedback o Emergency safing condition model	o Emergency state condition	o Last commanded state in s/s's command table	o Decomp. of state commands o Commands to subsystems	STATES
o Reporting functions of subsystems				o s/s's of TELECOM, POWER, THERMAL, MOBILITY, SCIENCE, SAMPLING	WORLD

FIGURE 5-3

TELECOMMUNICATIONS

SENSOR PRSING.	(SENSOR) WORLD MODEL	GLOBAL DATA B.	(STATE) WORLD MODEL	TASK DECOMP.	
					TASK PLANNING
		o Transmission point in inertia space o Vehicle location in inertial space	o Transmission models o Transformations	o Timing and transmission sequence o Direction in inertial coordinates	SEQUENCE OF PERMISSIBLE STATES
o Decoded signal o Location of antenna in vehicle coordinates	o Decoding scheme	o Data for/ from transmission o Vehicle location	o Encoding scheme o Transforms	o Direction for pointing in veh. coord. o Trans. to servo cmds. o Transmitter configuration o Coded data	PERMISSIBLE STATES
o Decommutation signal o Change in encoder counts	o Decommutation scheme o Readouts for feedback to motors and antenna configuration	o En/Decoded data for transmission o Servo state condition	o Commutation scheme o Servo gains o Configuration model	o Commutation signals for transmissions o Servo motor control o Transmitter configuration commands	STATES
o Encoders/ motors o Receivers				o Antennas and motors o TWT's and transmitters o Transponders	WORLD

FIGURE 5-4

P O W E R

SENSOR PRSING.	(SENSOR) WORLD MODEL	GLOBAL DATA B.	(STATE) WORLD MODEL	TASK DECOMP.	
					TASK PLANNING
		o Load state changes	o Load transition models	o Load transition commands	SEQUENCE OF PERMISSIBLE STATES
		o Load commanded state and last state achieved	o Load state o Load/ vehicle state model	o Load calculation o Load balance commands	PERMISSIBLE STATES
o Open/close junction state o State of discharge/ charge	o Feedback for control loops	o Charge/ discharge state o Array servo state	o Charge/ Discharge model o Distribution model o Servo commands	o Network distribution switching commands o Charge/ Discharge control loop o Servo control	STATES
o Voltmeters o Ampmeters o Network junctions				o Power generation: RTG's, Arrays o Power storage: batteries o Power distribution network	WORLD

FIGURE 5-5

T H E R M A L

SENSOR PRSING.	(SENSOR) WORLD MODEL	GLOBAL DATA B.	(STATE) WORLD MODEL	TASK DECOMP.	
					TASK PLANNING
					SEQUENCE OF PERMISSIBLE STATES
					PERMISSIBLE STATES
o Heater state o Temperature o Change in counts	o Temperature feedback	o Heater commands	o Heater state model o Vehicle thermal model	o Heating control loop o Heater state commands	STATES
o Rheostats o Thermostats o Encoder on louvers				o Louvers o Heaters	WORLD

FIGURE 5-6

SCIENCE

SENSOR PRSING.	(SENSOR) WORLD MODEL	GLOBAL DATA B.	(STATE) WORLD MODEL	TASK DECOMP.	
					TASK PLANNING
o Features correlated with expect. o Locations of inst. in inertial coordinates	o Expectation models: geology form. mineral content	o Terrain-map-based features of interest o Vehicle inertial location	o Sampling or science constraint/ locales o Transforms	o Planning of science imaging for sampling o Inertial coord. directions	SEQUENCE OF PERMISSIBLE STATES
o Science/ Sampling features extracted o Locations of inst. in vehicle coord.	o Geometric and spectral processing parameters o Feedback of inst. pos. in vehicle coordinates	o Vehicle location o Commanded state	o Transforms	o Directions for imaging o Coordinate transforms o Processing into state commands	PERMISSIBLE STATES
o Spectral and signal processing o Digitized imagery o Change in encoder counts	o Sounder parameters o Spectral band param. o Feedback for servos	o Servo state condition	o Last servo cmd. o Servo gains	o Servo command o Servo/motor control law	STATES
o Encoders for the motors o Imaging Spectrometer o Sounder/ Antenna				o Motors for Imaging Spectrometer scan platform o Motors for Sounder, Antenna articulation	WORLD

FIGURE 5-7

M O B I L I T Y

SENSOR PRSING.	(SENSOR) WORLD MODEL	GLOBAL DATA BASE	(STATE) WORLD MODEL	TASK DECOMP.	
o map for use in correlations		o route commanded	o expecta-tions model o sampling objectives o simplified vehicle models	o updated expectations o path planning o correlation of features with constraints o local path in the terrain	TASK PLANNING
o feature determination o vehicle state progress	o vehicle state model	o fused terrain map o commanded local path	o simplified vehicle model with con-straints o last cmd.'d path o obstacle model	o way points of interest to achieve o obstacle detection/avoid o local path calculation and commands	SEQUENCE OF PERMISSIBLE STATES
o feature map o vehicle inclination and heading o ground speed o distance traversed	o sensed state of vehicle and terrain o sum.'d/ diff.'d readouts	o commanded vehicle state o readouts from sensors	o commanded states o vehicle kinematics and dynamics	o commanded turns, moves up/down o coordinated commands to wheels and steering	PERMISSIBLE STATES
o digitized local map o range cnts. o angular cnts. o abs./rel. change in position in veh. coord.	o range reads o camera models o encoder, gyro readouts o digitized imagery	o servo state cmds	o servo gains o gear state o last cmd.'d servo state	o servo commands o servo/motor control laws	STATES
o Encoders o Cameras o Lasers o Inclinometers o Gyros/IMU o Accelerometers o Satellite cameras o Odometers/ speed sensors				o wheel motors o gears o drive train o steering motors o motors for pan/tilt head	WORLD

FIGURE 5-8

SAMPLING

SENSOR PRSING.	(SENSOR) WORLD MODEL	GLOBAL DATA B.	(STATE) WORLD MODEL	TASK DECOMP.	
o Fusion of features with terrain	o Task error models o Local terrain models o Feedback of sequence states	o Sampling site maps o Commanded sequences	o Simplified kinematic models o Spatial models of sampling sites o Resource models	o Planning of sampling task o Commanded sequences of grasps, manipulations and placement	TASK PLANNING
o Fusion of position, force with data base of objects sampled	o Sampling error models o Sensor-based feedback	o Manipulation sequence o Sampling states, site data	o Kinematics and dynamics of sets of links, end-eff. o Sampling site data base of positions and sizes	o Inertial coordinate transforms o Collision detection/ avoidance o Cmd.'d pos. forces, way pts.	SEQUENCE OF PERMISSIBLE STATES
o Ranges o Edges, Vert. centroids o Features extracted o Positions, forces in cart. coords., of veh	o Position and force control loop feedback	o Commanded state of links and end-effectors	o Kinematics and dynamics of sets of links o Tool and end-effector sizes/char. models	o Reference coordinate transformation o Trajectory calculations o Coordinated link/motor/end-effector cmds.	PERMISSIBLE STATES
o Stereo correlation o Digitized images and sensor readouts o Change in counts of pos. rate, range, acceleration	o Servo loop feedback	o Servo state condition	o Last servo state o Servo gains	o Servo cmd. o Servo/ motor control	STATES
o Encoders o Sensors of force/torque position, proximity o Cameras o Lasers				o Mechanical links o Motors o Tools and end-effectors: sampling devices	WORLD

FIGURE 5-9

The mobility block is shown in Figure 5-8. The highest level is focused on the task of developing then utilizing a local terrain map in route planning as the discussed under Section 3. After a 30 meter planned path is selected, the vehicle can begin to roll. The rover executes this 30 meters 10 meters at a time. The execution of this 10 meter traverse is detailed below.

The presented example shows the interaction between the blocks as well as control loops within the layers of given blocks. As we mentioned before, sampling is not presented for the sake of brevity. However, it should be noted that a sampling example was easily constructed since sampling is (assumed) implemented using telerobotics and the basic model was derived from NASREM, an architecture for telerobotics.

6. MOBILITY

In this section we will give an example of an operation of the rover vehicle and the accommodation provided by the functional architecture. In this operation, a command has been received by the rover from the mission team to move to a new location, based on the transmitted views of the site. The rover system must evaluate the surrounding terrain and generate a detailed route which reaches the commanded location. In arriving at a decision that the route so generated is suitable for traverse, the rover system must evaluate its state and determine on-board the feasibility of execution of the route. If so feasible, the rover begins the traverse. Periodic evaluation of progress to the goal state (i.e., the commanded location) during the traverse allows the rover system to determine the end of the operation as well as to proceed within available resources and within limits of safe operation.

In summary fashion the steps executed by the rover system in accomplishing the traverse include:
(1) tasks performed by the ground support team in determining the new location for the rover
(2) the up-link and
(3) on-board processing of command sequences which result in receipt by the rover of new goal state
(4) the determination of a feasible route to the goal state by the rover system
(5) the planning necessary to determine a route
(6) the execution of a traverse along the planned route.

In performing these steps several subsystems within the architecture interact to execute the required functions. Particular capabilities of these subsystems in accomplishing these functions are identified in summary fashion:

(1) EXECUTIVE: sequencing of subsystem support in the determination of a feasible route;
collection/evaluation of periodic reports of progress during execution of the traverse
final acceptance of reaching of the goal state

(2) TELECOMMUNICATION:
commutation and de-commutation of commands and telemetry

(3) POWER: determination of available power resources in support to route planning

(4) MOBILITY: provide imaging data of the near vicinity of the rover;
perform route planning;
execute rover movements which effect the traverse along the route

(5) SCIENCE: provide instrument data products which support evaluation of the site

The following is a detailed discussion of the example of execution of a traverse. The flow of activity follows the 'exploded' pictorial representation of the subsystems given in Figures 6-1 to 6-4 . Interspersed in the discussion is bracketed ([...]) references to specific steps shown as a flow of activity in the architecture on these figures.

6.1 GOAL DEFINITION

The mission science team evaluates the latest data concerning the geological and mineralogical properties of the site surrounding the rover. This data is a compilation of over-flight imaging taken by the companion orbiter, data available from past missions (e.g., Viking), observations by the on-board rover science instruments (possibly an imaging spectrometer and sounder), and range and feature data provided by the imaging components of the on-board rover mobility system. A new location (or set of locations) of interest is selected and registered as a (set of) mission objective. In this example a location 10m from the current site of the vehicle is identified [1a of Figure 6-1].

The vehicle team in mission control evaluates the selected locations based on models of (recent) past rover performance. An evaluation of feasibility in terms of vehicle health state and resource availability is performed [1b of Figure 6-1]. Simulations of the vehicle traverse over the terrain (available from over-flight and on-board rover imaging) assist in providing the feasibility check [interactive loop of Figure 6-1 between world model and task decomposition at the Mission Planning level of the executive]. A recommendation is provided to a mission director [the mission planner of task decomposition at the Mission Planning level of Figure 6-1], who in turn weighs the science return/objective(s) against the vehicle utilization. Conflicts (if any) are resolved and a new location (or set of locations) are identified for the rover [1c of Figure 6-1].

6.2 UP-LINK

The new location (in an appropriate inertial frame) along with related route information, including the expected length of the traverse (time, distance) and points of interest along the route (for collateral imaging and science instrument observation), is passed to Telecommunications for encoding [2a of Figure 6-2], commutation and up-link transmission [2b of Figure 6-2]. [N.B., These steps are performed by a separate though similar Telecommunications subsystem located on earth. For purposes of illustration these steps are shown on Figure 6-2.]

In addition, appropriate support data is commanded to other systems which will provide the latest update to the rover. This includes a topographic map generated as a result of over-flight by the planetary orbiter and a new reference location in inertial space for the rover developed by interferometry from tracking data.

DATA MANAGEMENT (EXECUTIVE)

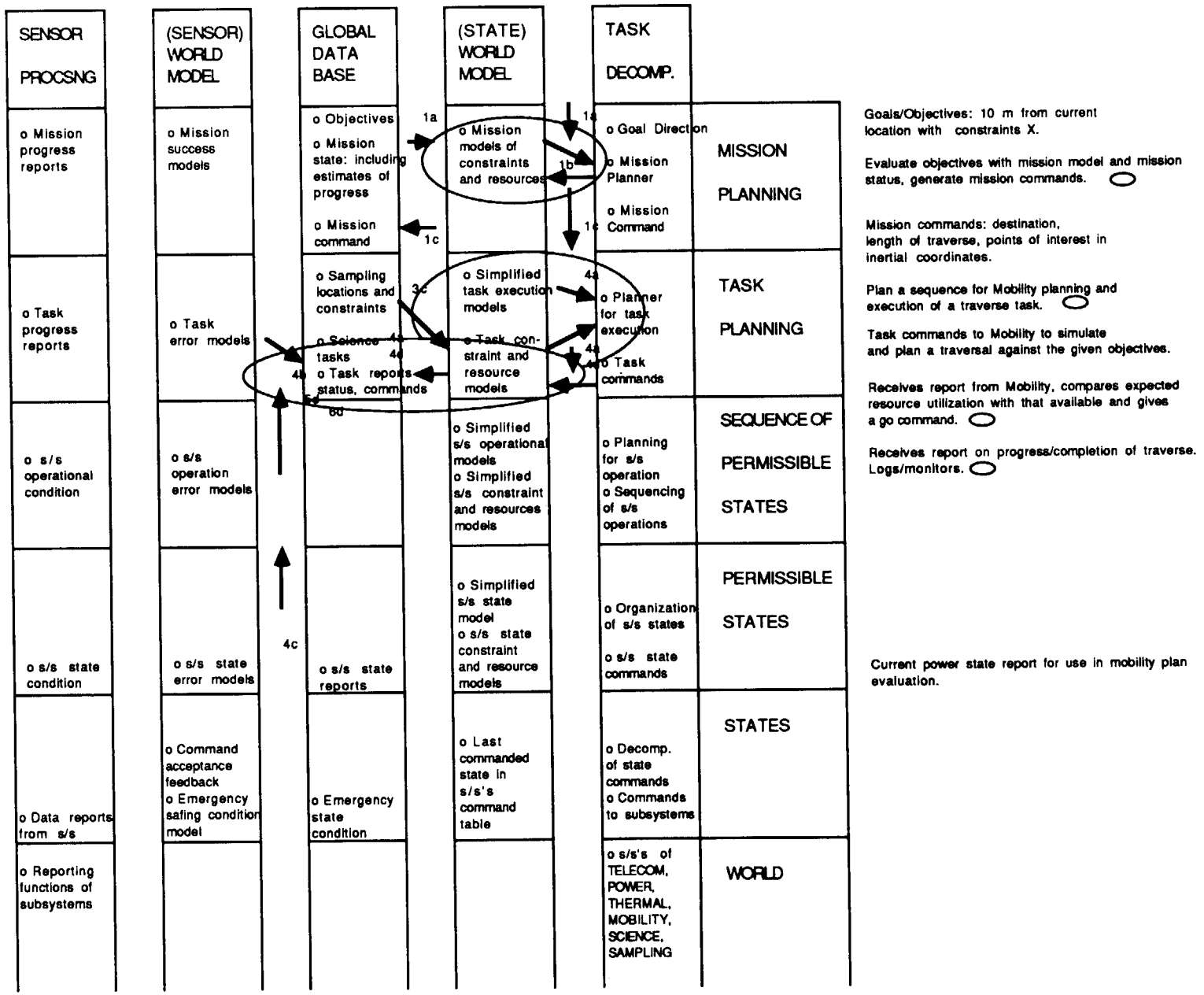


FIGURE 6-1

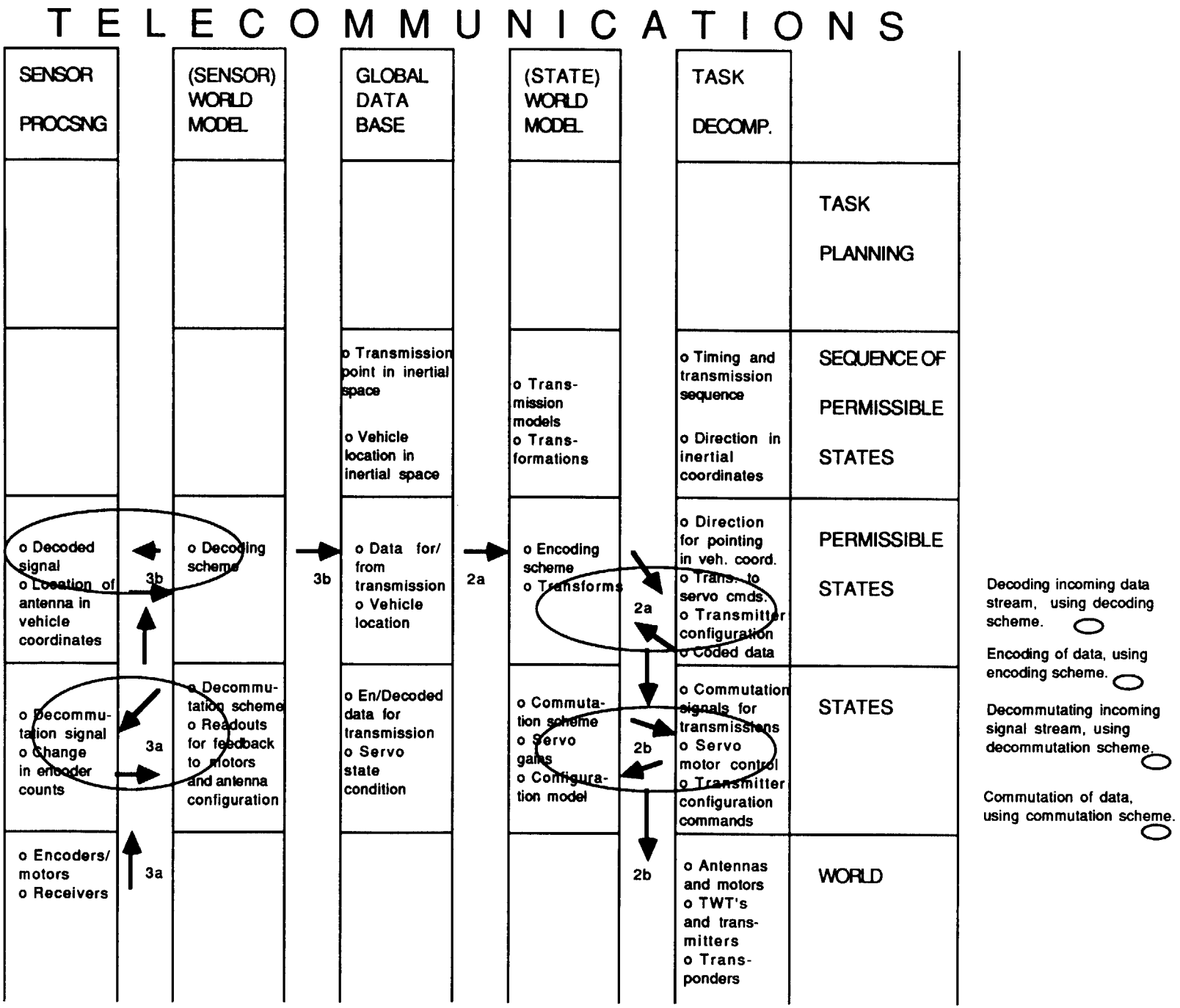

TELECOMMUNICATIONS
SENSOR PROCSNG
(SENSOR) WORLD MODEL
GLOBAL DATA BASE
(STATE) WORLD MODEL
TASK DECOMP.
TASK PLANNING
SEQUENCE OF PERMISSIBLE STATES
PERMISSIBLE STATES
STATES
WORLD
o Transmission point in inertial space
o Vehicle location in inertial space
o Trans-mission models
o Trans-formations
o Timing and transmission sequence
o Direction in inertial coordinates
o Decoded signal
o Location of antenna in vehicle coordinates
o Decoding scheme
o Data for/ from transmission
o Vehicle location
o Encoding scheme
o Transforms
o Direction for pointing in veh. coord.
o Trans. to servo cmds.
o Transmitter configuration
o Coded data
o Decommu-tation signal
o Change in encoder counts
o Decommu-tation scheme
o Readouts for feedback to motors and antenna configuration
o En/Decoded data for transmission
o Servo state condition
o Commuta-tion scheme
o Servo gains
o Configura-tion model
o Commutation signals for transmissions
o Servo motor control
o Transmitter configuration commands
o Encoders/ motors
o Receivers
o Antennas and motors
o TWT's and trans-mitters
o Trans-ponders
3a
3b
2a
2b
Decoding incoming data stream, using decoding scheme.
Encoding of data, using encoding scheme.
Decommutating incoming signal stream, using decommutation scheme.
Commutation of data, using commutation scheme.

FIGURE 6-2

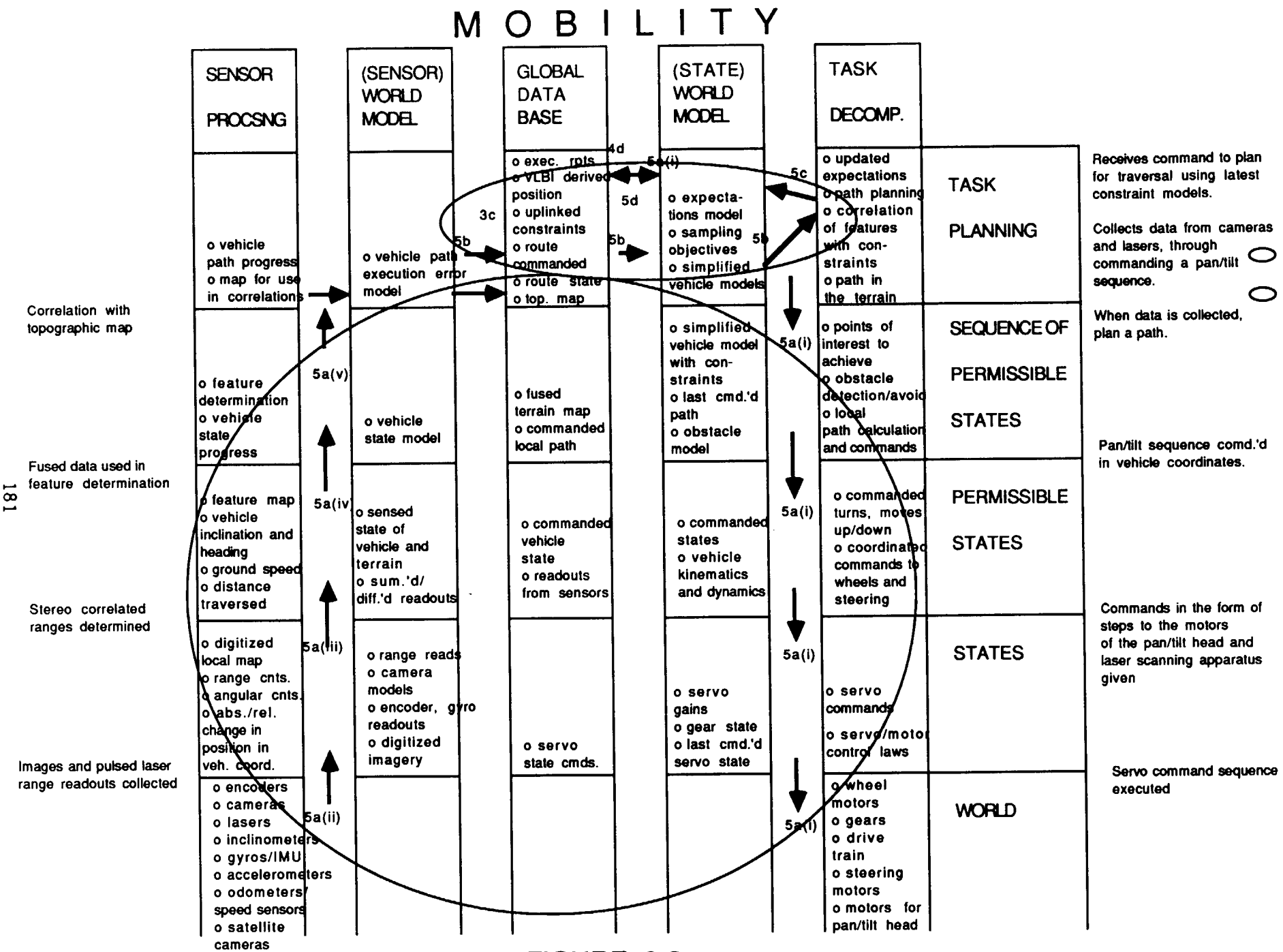
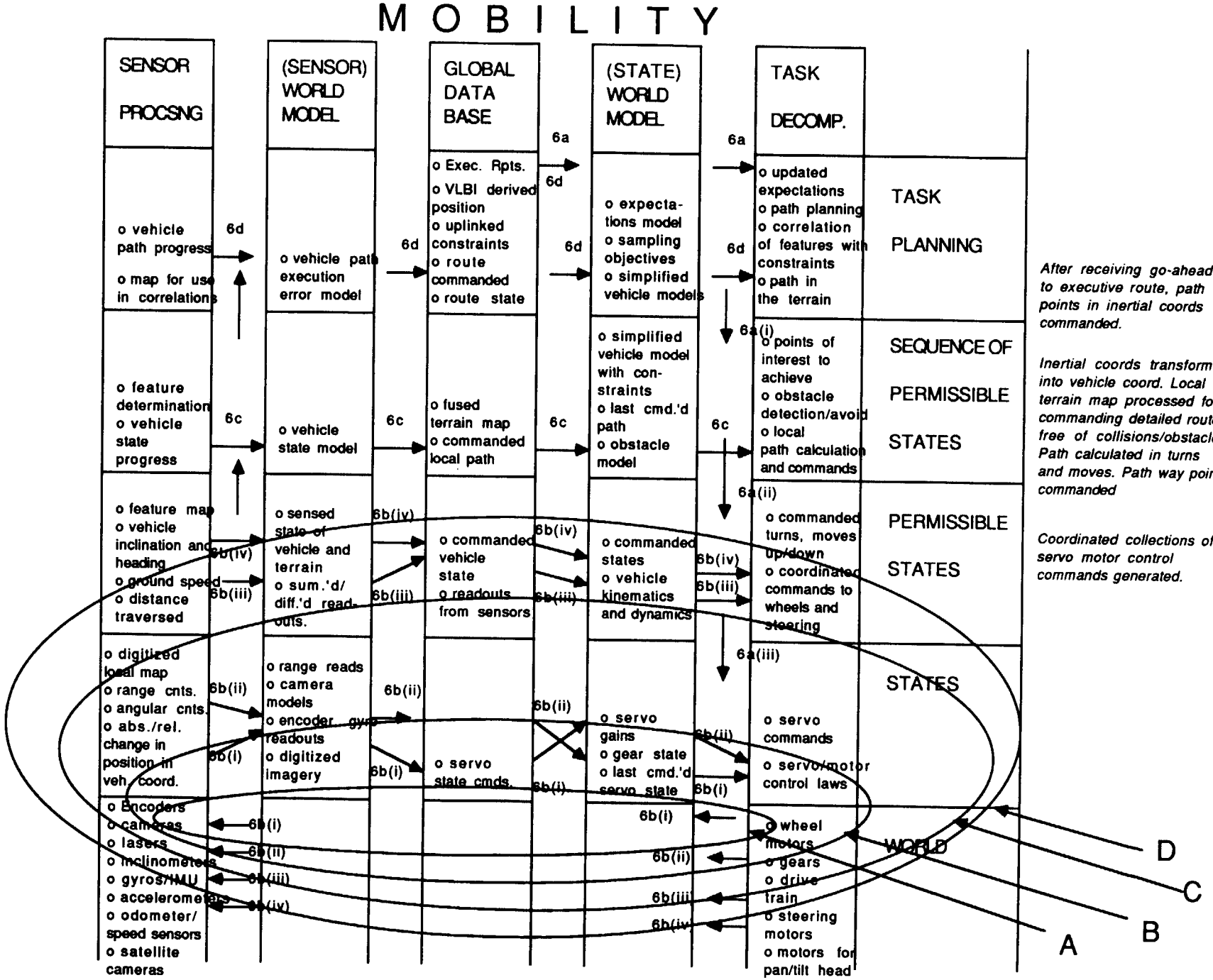

MOBILITY
SENSOR PROCSNG
(SENSOR) WORLD MODEL
GLOBAL DATA BASE
(STATE) WORLD MODEL
TASK DECOMP.
o vehicle path progress
o map for use in correlations
o vehicle path execution error model
o exec. rpts
o VLBI derived position
o uplinked constraints
o route commanded
o route state
o top. map
o expecta-tions model
o sampling objectives
o simplified vehicle models
o updated expectations
o path planning
o correlation of features with con-straints
o path in the terrain
TASK PLANNING
Receives command to plan for traversal using latest constraint models.
Collects data from cameras and lasers, through commanding a pan/tilt sequence.
Correlation with topographic map
o feature determination
o vehicle state progress
o vehicle state model
o fused terrain map
o commanded local path
o simplified vehicle model with con-straints
o last cmd.'d path
o obstacle model
o points of interest to achieve
o obstacle detection/avoid
o local path calculation and commands
SEQUENCE OF PERMISSIBLE STATES
When data is collected, plan a path.
Fused data used in feature determination
o feature map
o vehicle inclination and heading
o ground speed
o distance traversed
o sensed state of vehicle and terrain
o sum.'d/ diff.'d readouts
o commanded vehicle state
o readouts from sensors
o commanded states
o vehicle kinematics and dynamics
o commanded turns, motes up/down
o coordinated commands to wheels and steering
PERMISSIBLE STATES
Pan/tilt sequence comd.'d in vehicle coordinates.
Stereo correlated ranges determined
o digitized local map
o range cnts.
o angular cnts.
o abs./rel. change in position in veh. coord.
o range reads
o camera models
o encoder, gyro readouts
o digitized imagery
o servo state cmds.
o servo gains
o gear state
o last cmd.'d servo state
o servo commands
o servo/motor control laws
STATES
Commands in the form of steps to the motors of the pan/tilt head and laser scanning apparatus given
Images and pulsed laser range readouts collected
o encoders
o cameras
o lasers
o inclinometers
o gyros/IMU
o accelerometers
o odometers/ speed sensors
o satellite cameras
o wheel motors
o gears
o drive train
o steering motors
o motors for pan/tilt head
WORLD
Servo command sequence executed
5a(i)
5a(ii)
5a(iii)
5a(iv)
5a(v)
5b
5c
3c
4d
5d

FIGURE 6-3



After receiving go-ahead to executive route, path points in inertial coords commanded.

Inertial coords transformed into vehicle coord. Local terrain map processed for commanding detailed route free of collisions/obstacles. Path calculated in turns and moves. Path way points commanded

Coordinated collections of servo motor control commands generated.

FIGURE 6-4

6.3 RECEIVE COMMAND

The commanded position and associated data are received through Telecommunications. The data is de-commutated [3a of Figure 6-2], decoded [3b of Figure 6-2] and stored [through the use of the global data base connecting the subsystems] for use in subsequent operations [3c in Figure 6-1 for the command and associated constraints, and 3c in Figure 6-3 for the topographic map and reference location].

6.4 ROUTE PLANNING

The receipt of a command for a traverse to a new location at the Executive [3c of Figure 6-1] begins a planning activity which leads to the generation of a feasible route. The executive calls upon the subsystems [4a of Figure 6-1] to report the current state of resources available for the traverse [4c of Figure 6-1]. These status reports become a part of the constraint space used by the route planning function [4a of Figure 6-1]. A major portion of this function is located in the Mobility subsystem [Figure 6-3]. Here the current state data gathered from the imaging systems on-board the rover are evaluated against the models of the terrain to determine the route to the new location (see 6.4.1 below). Once the route is generated (with an expected resource utilization) a final go/no go evaluation is performed by the executive [4d of Figure 6-1] using the data in any subsystem or other reports received during route planning. A 'go' from the executive to Mobility begins the traverse to the new location.

6.4.1 MOBILITY: ROUTE PLANNING

Route planning begins with the gathering of the latest imaging data of the site surrounding the present position of the rover. A panoramic view of the site is developed through the execution of a sequence of moves of (for example) a pan and tilt mounted camera system on the rover. The precise sequence, based on the current vehicle state, is developed (at the task planning level of Mobility) and commanded for execution [5a(i) of Figure 6-3]. The commands, initially in inertial coordinates, are transformed into specific motor drive commands to the motor mechanisms supporting the camera system [the hierarchical flow of commands through the levels of the Mobility subsystem, labelled 5a(i)]. At each position in the sequence, an image pair (for stereo) is captured and digitized for further processing [5a(ii) of Figure 6-3]. The application of camera models, correlation of multiple images and image processing result in developing range and dimension of specific features [5a(iii) of Figure 6-3]. A correlation of image features to models of terrain features (e.g., boulders, ravines) across several image pairs results in a list of position- and dimension-registered objects (or obstacles) for use in route planning [5a(iv) of Figure 6-3]. A final correlation to an on-board topographic map allows computation of rover position in the terrain and a selection (based on the commanded new position) of the portion of the map for use by the route planner [5a(v) of Figure 6-3]. These results are stored [in the global data base] in the form of a fused map of the route locale of greater resolution than available from the up-linked topographic map alone for use in subsequent route planning.

The route planner generates a path to the goal state (the commanded location of the rover) which satisfies the intermediate view point criteria (if any), the vehicle physical constraints (e.g., clearance, power utilization) and local obstacle avoidance [5b of Figure 6-3]. As part of the verification of the suitability of the path, the movement of the vehicle in the terrain is simulated. In addition, during the simulation expectation models of vehicle performance are generated for use in monitoring the actual traverse of the rover along the path [5c of Figure 6-3]. Only a portion of the planned path is slated for execution, as the errors in planning increase as a function of range. In the case of goal of 10m nominally the entire path can be executed. A report to this effect is generated for final go/no go disposition by the executive [5d of Figure 6-3].

6.5 MOBILITY: PATH TRAVERSE

The planned path is executed in Mobility upon receiving a go ahead command [6a of Figure 6-4]. The planned path in inertial coordinates is transformed into vehicle coordinates [6a(i) of Figure 6-4], specific coordinated moves and turns by the vehicle carriage [6a(ii) of Figure 6-4] and servo commands to the motors [6a(iii) of Figure 6-4].

Each command type represents the output command of different type of control law used in the execution of vehicle motion. At the lowest level [State Level of Mobility] a loop is closed around the servo command using the feedback from motor encoders [6b(i) and the innermost loop A in Figure 6-4]. At the next level a heading/dead reckoning loop is closed around the turn or move commands as measured through the feedback of gyros/IMU or compass [6b(ii) and loop B in Figure 6-4]. A control loop around the distance traversed command is closed through feedback measurements from accelerometers and over-the-ground sensor [6b(iii) and the loop C in Figure 6-4]. A final control loop is centered around the constraint of providing a stable platform during the traverse. The feedback for this loop is provided by inclinometer and integration of the readouts of gyros/IMU [6b(iv) and loop D of Figure 6-4]. Each loop stores feedback data, commands and state estimates for evaluation of vehicle performance during the traverse.

Other processing during the traverse monitors and performs the vehicle state evaluation based on the progress reported by lower levels of the hierarchy. In particular, the progress of the traverse is compared against the model of performance along the path. Corrections in the form of commanded turns and moves ensure compliance to way point and obstacle avoidance constraints [6c of Figure 6-4]. Lastly, progress to the goal state is monitored with periodic reports issued to the executive [6d of Figure 6-4 and Figure 6-1]. A final report of reaching the new location ends the traverse and monitoring loop in the executive.

7. UTILITY AND APPLICATION

The architecture as it is laid out forms the basis of a complete Rover Architecture. When completed it should contain all of the functions both on the ground and in flight. It is at once a complete description of both the control and information flows. Step by step sequences and loops can be worked out for various strategies so the designer can see the interface complexities directly.

Evaluating the rate of execution of these loops can aid in identifying technology alternatives to achieve a greater mission capability. As was discussed at the end of section 4, execution rate is one factor in determining where functions sit in the architecture. If a mission planner wishes to increase the range of the rover, more functions must be 'pushed further down' in the hierarchy or greater processing technology brought to bear to achieve the required performance.

In addition to basic design strategies, detailed fault protection scenarios can be worked out, making sure control loops are continuous and unbroken. Often, in fault protection design, the required elements of the design are difficult to identify. This architecture shows that a fault protection machine must be considered for each state machine in each subsystem. The architecture aids in identifying communication paths (commands and telemetry) needed to achieve the fault protection capability. Tracing scenarios of recovery (as was done in section 5 in brief) reveal the communication paths required.

Finally, by computerizing this model, representing each state machine by the convention $M^k_{ij}(l)$ for each element and including these elements in a data base, control loops can be easily described by strings of execution of elements. Strings repeatedly used in these control loops can be identified and represent the state machines which must be developed and tested first.

8. CONCLUSION

A general architectural concept for a planetary rover is presented, based on an expansion of the NASREM concept for telerobotic control. A mapping of this architecture with the functions for a rover in a MRSR mission has been performed. An example of a mobility scenario executed within this architecture validates the concept. This example and associated discussion illustrate the capability of the architecture to serve as a tool for design and functional trade-off analysis.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the Office of Aeronautics and Space Technology, National Aeronautics and Space Administration.

REFERENCES

1. Wilcox, B.H., Gennery, D.B., Mishkin, A.H., "Mars Rover Local Navigation and Hazard Avoidance," Mobile Robots III, Proceedings of SPIE (to be published).

2. Brooks, R., "A Robot That Walks; Emergent Behaviors from a Carefully Evolved Network," Computation, Volume 1, December 1, Spring, 1989.

3. "Carnegie-Mellon Mars Rover", NASA Review held October 14, 1988.

4. "Mars Rover/Sample Return (MRSR) Rover Mobility and Surface Rendezvous Studies", Final Report, JPL Contract 958073, Martin Marietta Space Systems Company, Denver, Colorado, July, 1988.

5. Albus, J.S., McCain, W.G., Lumia, R., "NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)," National Bureau of Standards Robot Systems Division, 12/4/86.

6. Kovtunenko, V.M., Kemudzian, A.L., Sukhanov, K.G., "Mars Rover: Fundamental Control Principles," Babakin Engineering and Research Centre, IAF-88-397.

7. Dias, "Revised Rover Surface Operations Scenarios Baseline," JPL IOM (internal document), dated 29 June 1988.

8. "Robotic Vehicle Computer Aided Remote Driving", JPL D-3282 (internal document), June, 1986.

9. Kwok, J.H., "MRSR Reference Missions Summary," Version 23. (internal project document), September 14, 1988.

10. "Final Report: Flight Telerobotic Servicer (FTS) Tinman Concept; In-House Phase B Study", SS-GSFC-0042, Volumes I, II, Goddard Space Flight Center, September 9, 1988.

11. "Functional Requirements for the Telerobotic Testbed," JPL D-3693 (internal document), Revision 3 December, 1988.